

TW

**stichting  
mathematisch  
centrum**



AFDELING TOEGEPASTE WISKUNDE

TN 59/71

MEI

TW

M.J.W. JANSEN

SEMI-ITERATIEVE METHODEN VOOR DE OPLOSSING  
VAN DE VERGELIJKING VAN LAPLACE

**2e boerhaavestraat 49 amsterdam**

BIBLIOTHEEK MATHEMATISCH CENTRUM  
AMSTERDAM

*Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.*

*The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.*

Inhoud

	pag.
0. Inleiding	2
1. Probleemstelling	2
2. Minimalisatie van de fout	5
2.1. Eigenwaarden van H	5
2.2. Keuze van een norm	5
2.3. Benadering van gekozen normen	6
2.4. Convergentiesnelheid	7
3. Vorm-minimaliserende polynomen	7
3.1. Maximum-norm (Chebyshev-polynomen)	8
3.2. Gewogen twee-norm (kern-polynomen)	8
4. Toepassingen	10
4.1. Proces I	10
4.2. Proces II	11
5. Waarnemingen	12
5.1. Convergentiesnelheden	12
5.2. Effect van de gewichtsfunctie	17
5.3. Conclusies	18
6. ALGOL 60 programma voor proces II	18
Literatuur	22

## 0. Inleiding

De vectorvergelijking die ontstaat bij passende discretisatie van het Dirichlet-probleem voor de Laplace-vergelijking kan worden opgelost met behulp van het semi-iteratieve proces:

$$u_k = P_k(H) u_0.$$

Hierin is  $u_0$  een beginvector die op de randen van een gebied voorgescreven waarden aanneemt;  $u_k$  een benadering van de gezochte oplossingsvector;  $H$  een matrix waarvan vele elementen nul blijken te zijn (een "sparse matrix"); en  $P_k$  een polynoom dat iteratief wordt geconstrueerd.

Het polynoom  $P_k$  moet zó geconstrueerd worden, dat de rij  $u_k$  zo snel mogelijk convergeert. Om dit doel te bereiken kiest men voor  $P_k$  meestal een Chebyshev-polynoom, Stiefel [2] ziet een mogelijkheid de convergentie van  $u_k$  te versnellen door  $P_k$  zó te kiezen, dat bij het iteratie-proces speciale nadruk komt te liggen op die gebieden van het spectrum van  $H$ , waar de beginfout geacht wordt het grootst te zijn.

In dit verslag worden theorie en experiment vergeleken.

## 1. Probleemstelling

Het Dirichlet-probleem voor de Laplace-vergelijking op een vierkant, leidt tot het volgende gediscretiseerde probleem (zie [1]):

Gevraagd een functie  $u$  die voldoet aan

$$(1.1) \quad (1-H) u = 0 \quad \text{op } B,$$

$$(1.2) \quad u = f \quad \text{op } \partial B.$$

Hierin is

$u$  een roosterfunctie (vector) die waarden aanneemt op het rooster  $\bar{B}$ :

$$\bar{B} = \{(rx, ry) \mid rx, ry = 0, 1, 2, \dots, R\},$$

B de verzameling inwendige punten van  $\bar{B}$ :

$$B = \{(rx, ry) \mid rx, ry = 1, 2, \dots, R-1\},$$

en  $\partial B$  de verzameling randpunten van B:

$$\partial B = \bar{B} - B.$$

De operator H, die voorlopig alleen op het binnengebied B gedefinieerd is, berekent het gemiddelde van u over de vier naaste burens van  $(rx, ry)$  i.e.

$$(1.3) \quad Hu|_{(rx, ry)} = \frac{1}{4} [u(rx, ry-1) + u(rx-1, ry) + u(rx+1, ry) + u(rx, ry+1)].$$

Men gebruikt hiervoor vaak de molecuul-notatie:

$$(1.4) \quad \text{mol}(H) = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Verder voeren wij in:

$G$ , de verzameling van functies op  $\bar{B}$ , en

$G_0$ , de verzameling van functies op  $\bar{B}$  die op  $\partial B$  nul worden.

We zullen een rij  $\{u_k\}$  genereren, die moet convergeren naar u. Aangezien we de waarden van u op  $\partial B$  kennen, kunnen we er voor zorgen dat de fout

$$(1.5) \quad v_k = u_k - u,$$

in de ruimte  $G_0$  ligt.

Stel dat

$$(1.6) \quad v_k = P_k(H) v_0, \quad v_0, v_k \in G_0,$$

waarin  $P_k$  een polynoom van de graad  $k$  is; dan geldt voor  $u_k$ :

$$(1.7) \quad u_k = P_k(H) u_0 + (1 - P_k(H)) u.$$

We kunnen  $H$  op de rand nog vrij kiezen, en nemen:

$$(1.3') \quad Hu|_{(rx,ry)} = u(rx,ry), \quad \text{op } \partial B;$$

of in molekuulnotatie:

$$(1.4') \quad \text{mol}(H) = [1], \quad \text{op } \partial B.$$

Hiermee is (1.1) uitgebreid tot het hele gebied  $\bar{B}$ :

$$(1.1') \quad (1-H) u = 0, \quad \text{op } \bar{B}.$$

Indien we  $P_k$  zó normeren, dat

$$(1.8) \quad P_k(1) = 1,$$

reduceert (1.7) tot

$$(1.9) \quad u_k = P_k(H) u_0.$$

In verband met (1.6) moet

$$(1.10) \quad u_0 = f \quad \text{op } \partial B.$$

## 2. Minimalisatie van de fout

### 2.1. Eigenwaarden van H

In verband met de afschatting van de fout,  $v_k = P_k(H) v_0$ , is het volgende van belang (zie [1]): De eigenvectoren  $e_j$ , van H, vormen een compleet, orthogonaal stelsel in  $G_0$ , en de eigenwaarden  $\mu_j$  liggen tussen de grenzen

$$(2.1) \quad \text{mumin} \leq \mu_j \leq \text{mumax},$$

waarin

$$\text{mumin} = -1 + \frac{1}{2} h^2,$$

$$\text{mumax} = 1 - \frac{1}{2} h^2,$$

$$h = \pi/R \quad (\text{roosterafstand}).$$

We zullen aannemen dat de eigenvectoren  $e_j$ ,  $j = 1, 2, \dots, (R-1)^2$ , gerangschikt zijn naar opklimmende waarden van  $\mu_j$ , en dat  $(e_j, e_j) = 1$ .

### 2.2. Keuze van een norm.

De beginfout  $v_0$  kan ontbonden worden in eigenvectoren van H:

$$(2.2) \quad v_0 = \sum c_j e_j.$$

Voor  $v_k$  vindt men dan

$$(2.3) \quad v_k = P_k(H) v_0 = \sum c_j P_k(\mu_j) e_j,$$

zodat

$$(2.4) \quad ||v_k||_2^2 = \sum c_j^2 P_k^2(\mu_j) \leq ||P_k||_2^2 ||v_0||_2^2$$

en

$$(2.5) \quad ||v_k||_{\max} = \max_j |c_j P_k(\mu_j)| \leq ||P_k||_{\max} ||v_0||_{\max}.$$

Men kan  $V_k$  snel naar nul laten gaan door:

1° minimalisatie van  $||P_k||_{\max}$  (Richardson-Chebyshev-proces),

2° minimalisatie van  $||P_k||_2$ ;

beiden onder de nevenconditie

$$P_k(1) = 1.$$

Bij deze methodes wordt geen gebruik gemaakt van de mogelijkheid dat het gedrag van  $c_j$  globaal, maar niet exact, bekend kan zijn. Met behulp van een functie  $d_j$  die het globale gedrag van  $c_j$  vertoont, kan men pogen  $||V_k||_2$ , en niet  $||P_k||_2$ , te minimaliseren. Dit leidt tot een derde methode:

3° minimalisatie van  $\sum d_j^2 P_k^2(\mu_j) = ||P_k||_d^2$ ,  $d_j \neq 0$ .

Op deze wijze kan men delen van het spectrum van  $V_0$  bij het iteratieproces benadrukken (met het bijbehorende risico van verwaarlozing van het verdere spectrum).

### 2.3. Benadering van gekozen normen

Men kan de genoemde normen van  $P_k$  als volgt benaderen:

$$(2.6) \quad ||P_k||_{\max} \approx \max_{\text{mumin} < \mu < \text{mumax}} |P_k(\mu)|,$$

$$(2.7) \quad ||P_k||_2^2 \approx \int_{\text{mumin}}^{\text{mumax}} P_k^2(\mu) d\mu.$$

In formule (2.7) is geen rekening gehouden met het feit dat de eigenwaarden  $\mu_j$  onregelmatig over het interval  $(\text{mumin}, \text{mumax})$  verdeeld kunnen liggen; dit wordt hersteld bij de benadering van  $||P_k||_d^2$ :

$$(2.8) \quad ||P_k||_d^2 \approx \int_{\text{mumin}}^{\text{mumax}} P_k^2(\mu) g(\mu) d\mu,$$



waarin

$$\frac{1}{2}(\mu_{j+1} - \mu_{j-1}) g(\mu_j) \approx d_j^2.$$

#### 2.4. Convergentiesnelheid

In het algemeen zal de fout exponentieel naar nul gaan. We definiëren een feitelijke convergentiesnelheid

$$(2.9) \quad R^*(k) = -\frac{1}{k} \log \frac{||v_k||}{||v_0||},$$

en een theoretische convergentiesnelheid

$$(2.10) \quad R(k) = -\frac{1}{k} \log ||P_k||.$$

Aangezien

$$||v_k|| = ||P_k(H) v_0|| \leq ||P_k|| \cdot ||v_0||,$$

geldt

$$(2.11) \quad R^*(k) \geq R(k).$$

Het proces zal dus niet langzamer mogen convergeren dan theoretisch is voorspeld.

#### 3. Norm-minimaliserende polynomen

We voeren een transformatie uit die het interval  $(\mu_{\min}, \mu_{\max})$  lineair overvoert in het interval  $(-1, 1)$ :

$$(3.1) \quad y(\mu) = \frac{2\mu - (\mu_{\max} + \mu_{\min})}{\mu_{\max} - \mu_{\min}}.$$

Verder definiëren we:

$$(3.2) \quad \begin{cases} y_1 &= y(1) \quad , \quad (y_1 > 1), \\ Q_k(y) &= P_k(\mu) \quad , \quad y = y(\mu), \\ w(y) &= g(\mu) \quad , \quad y = y(\mu). \end{cases}$$

Het probleem is nu: de minimalisatie van

$$(3.3) \quad ||Q_k||_{\max} = \max_{-1 < y < 1} |Q_k(y)|,$$

en van

$$(3.4) \quad ||Q_k||_w^2 = \int_{-1}^1 Q_k^2(y) w(y) dy,$$

onder de nevenconditie

$$(3.5) \quad Q_k(y_1) = 1.$$

### 3.1. Maximum-norm (Chebyshev-polynomen)

Minimalisatie van de maximum-norm wordt gerealiseerd door

$$(3.6) \quad Q_k(y) = \frac{T_k(y)}{T_k(y_1)},$$

waarin  $T_k$  een  $k$ -de graads Chebyshev-polynoom van de eerste soort is (zie [1] en [3]).

De norm van  $Q_k$  is

$$(3.7) \quad ||Q_k||_{\max} = \{\cosh(k \log r)\}^{-1},$$

waarin

$$(3.8) \quad r = y_1 + \sqrt{y_1^2 - 1}.$$

Voor de theoretische convergentiesnelheid vindt men

$$(3.9) \quad R(k) = -\frac{1}{k} \log ||Q_k|| = \log r - \frac{1}{k} \log 2 + o\left(\frac{1}{k}\right).$$

### 3.2. Gewogen tweekern (kern-polynomen)

Zij  $\{q_k(y)\}$  een stelsel polynomen dat op het interval  $(-1, 1)$  orthogonaal is ten opzichte van de gewichtsfunctie  $w(y)$ :

$$(3.10) \quad \int_{-1}^1 q_k(y) q_l(y) w(y) dy = h_k \delta_{k,l}.$$

Beschouw nu het polynoom

$$(3.11) \quad K_n(y_1, y) = \sum_{j=0}^n h_j^{-1} q_j(y_1) q_j(y).$$

Polynomen van deze vorm noemt men kern-polynomen (zie bijvoorbeeld [2] en [3]); zij danken hun naam aan de eigenschap dat

$$(K_n(y_1, y), q_j(y))_w = q_j(y_1), \quad j \leq n.$$

Het kern-polynoom  $K_n$  is dus een reproducerende kern (een soort  $\delta$ -functie) voor willekeurige polynomen  $f_j$ , van graad  $j \leq n$ :

$$(3.12) \quad (K_n(y_1, y), f_j(y))_w = f_j(y_1), \quad j \leq n.$$

Voor de norm van  $K_n$  vinden we met behulp van (3.12) dat

$$(3.13) \quad \|K_n(y_1, y)\|_w^2 = K_n(y_1, y_1).$$

Het stelsel  $\{K_n(y_1, y)\}$  is orthogonaal ten opzichte van het gewicht  $(y_1 - y)w(y)$ ; uit (3.12) volgt immers voor een willekeurig polynoom  $f_j$ , van graad  $j < n$ , dat

$$(3.14) \quad \int_{-1}^1 K_n(y_1, y) f_j(y) (y_1 - y) w(y) dy = 0, \quad j < n.$$

De polynomen  $K_n$  zijn voor ons van nut wegens de volgende eigenschap (norm-minimalisatie):

Zij  $S_j$  een polynoom van graad  $j \leq n$ , met de standaardisatie

$$S_j(y_1) = K_n(y_1, y_1),$$

dan geldt

$$(3.15) \quad ||S_j(y)||_w \geq ||K_n(y_1, y)||_w.$$

Immers,

$$S_j(y) = K_n(y_1, y) + R_m(y),$$

waarin  $R_m$  een polynoom is van graad  $m \leq n$ , met

$$R_m(y_1) = 0;$$

we vinden voor de norm van  $S_j$ :

$$||S_j(y)||_w^2 = ||K_n(y_1, y)||_w^2 + ||R_m(y)||_w^2 + (K_n(y_1, y), R_m(y))_w;$$

wegens (3.12) is de laatste term nul; hieruit volgt de bewering.

Blijkbaar is

$$(3.16) \quad K_n(y_1, y)/K_n(y_1, y_1)$$

het door ons gezochte, norm-minimaliserende polynoom.

#### 4. Toepassingen

De resultaten van paragraaf 3.2 zijn door ons op twee manieren toegepast in een numeriek proces.

##### 4.1. Proces I

Minimalisatie van de norm

$$(4.1) \quad ||.||_{(1-y)^\alpha(1+y)^\beta},$$

wordt gerealiseerd door het polynoom

$$(4.2) \quad Q_n(y) = \frac{\sum_{j=1}^n h_j^{-1} q_j(y_1) q_j(y)}{\sum_{j=1}^n h_j^{-1} q_j(y_1) q_j(y_1)},$$

waarin  $q_j$  het Jacobi-polynoom  $P_j^{\alpha\beta}$  is.

Voor de convergentiesnelheid van dit proces in de norm (4.1) kan een asymptotische uitdrukking gevonden worden. De recursie-coëfficiënten  $a_n$ ,  $b_n$  en  $c_n$  van de Jacobi-polynomen  $P_n^{\alpha\beta}$  gaan naar de limieten 0, 2, respectievelijk 1. Hieruit volgt dat

$$q_n(y_1)/q_{n-1}(y_1) \rightarrow r, \quad \text{voor } n \rightarrow \infty;$$

waarin

$$(3.8) \quad r = y_1 + \sqrt{y_1^2 - 1}.$$

Met behulp van (2.10) en (3.13) vindt men na enig rekenen dat

$$(4.3) \quad R(n) \rightarrow \log r, \quad \text{voor } n \rightarrow \infty.$$

Blijkbaar is de convergentiesnelheid van dit proces in de limiet gelijk aan die van het Richardson-Chebyshev proces (3.9).

In het ALGOL-programma voor proces I is gebruik gemaakt van een 4-traps recursie voor de optimale polynomen  $Q_n$  (zie [1]). Wegens de orthogonaliteit (3.14) bestaat er voor het stelsel  $\{Q_n\}$  ook een 3-traps recurrente betrekking. In ons geval,  $q_j = P_j^{\alpha\beta}$ , kunnen de recursie-coëfficiënten numeriek bepaald worden. Het bijbehorende proces is volgens Stiefel [2] stabiel.

We zullen hier niet verder op dit proces ingaan, omdat de beschouwing van paragraaf 3.2 ons op het spoor brengt van een veel eenvoudiger proces, dat vergelijkbare resultaten kan opleveren.

#### 4.2. Proces II

Minimalisatie van de norm

$$(4.4) \quad || \cdot ||_{(1-y)^\alpha (1+y)^\beta (y_1-y)^{-1}}$$

wordt volgens (3.14) en (3.15) gerealiseerd door

$$(4.5) \quad Q_n(y) = P_n^{\alpha\beta}(y)/P_n^{\alpha\beta}(y_1).$$

Voor  $Q_n$  geldt de recurrente betrekking:

$$(4.6) \quad Q_{n+1}(y) = (a_n + b_n y) \frac{P_n^{\alpha\beta}(y_1)}{P_{n+1}^{\alpha\beta}(y_1)} Q_n(y) - c_n \frac{P_{n-1}^{\alpha\beta}(y_1)}{P_{n+1}^{\alpha\beta}(y_1)} Q_{n-1}(y),$$

waarin  $a_n$ ,  $b_n$  en  $c_n$  de recursiecoëfficiënten zijn van de Jacobi-polynomen  $P_n^{\alpha\beta}$ .

Het bijbehorende numerieke proces is stabiel [2]. De convergentiesnelheid is niet bekend. Het geval,  $\alpha = \beta = -.5$ , levert het Richardson-Chebyshev-proces.

## 5. Waarnemingen

### 5.1. Convergentiesnelheden

Voor enkele speciale gevallen berekenen we de convergentiesnelheid van het residu,

$$t_k = (1-H) u_k.$$

Deze is:

$$(5.1) \quad R_k = -\frac{1}{k} \log \frac{||t_k||}{||t_0||},$$

waarin als norm genomen is

$$||a|| = \max_{0 \leq rx, ry \leq R} |a(rx, ry)|.$$

Een streepje in de tabel duidt op een instabiel proces; voor het gebruikte stabiliteitscriterium zij verwezen naar [1]. Het aantal roosterpunten is bepaald door de keuze,  $R = 10$ .

Startvector 3.

$$u_0(rx,ry) = \sin(rx.h) \sin(ry.h)$$

Proces I

$\alpha$	$\beta$	R(10)	R(20)	R(50)
0	0	.15	.21	.26
0	1	.16	.22	.26
0	2	--	--	--
-.25	0	.17	.23	.27
-.5	0	.20	.26	.28
-.75	0	--	--	--

Proces II

-.5	-.5	.25	.28	.29
0	0	.19	.23	.26
0	1	.20	.24	.27
0	2	.21	.24	.27
0	3	.22	.25	.27
-.25	0	.21	.25	.28
-.5	0	.25	.28	.29
-.75	0	.32	.32	.30
.5	-.5	.14	.20	.25

Startvector 4

$$u_0(rx,ry) = (rx.h-2) (ry.h-2) \sin(rx.h) \sin(ry.h)$$

Proces I

$\alpha$	$\beta$	R(10)	R(20)	R(50)
0	0	.33	.31	.30
0	1	.33	.32	.30
0	2	--	--	--
-.25	0	.31	.31	.31
-.5	0	.29	.32	.31
-.75	0	--	--	--

Proces II

-.5	-.5	.26	.36	.31
0	0	.30	.33	.31
0	1	.30	.34	.31
0	2	.30	.36	.32
0	3	.31	.35	.32
-.25	0	.28	.35	.32
-.5	0	.27	.35	.31
-.75	0	.26	.31	.32
.5	-.5	.33	.30	.29



Startvector 5

$$u_0(rx,ry) = (rx.h-2) (ry.h-2) (rx.h-1) (ry.h-1) \sin(rx.h) \sin(ry.h)$$

Proces I

$\alpha$	$\beta$	R(10)	R(20)	R(50)
0	0	.48	.41	.34
0	1	.40	.39	.34
0	2	--	--	--
-.25	0	.39	.39	.33
-.5	0	.32	.35	.33
-.75	0	--	--	--

Proces II

-.5	-.5	.23	.30	.32
0	0	.31	.34	.33
0	1	.31	.33	.33
0	2	.36	.33	.29
0	3	.34	.26	.24
-.25	0	.27	.32	.33
-.5	0	.23	.29	.31
-.75	0	.20	.28	.31
.5	-.5	.47	.41	.34

Startvector 6

$$u_0(rx,ry) = \begin{cases} 1, & ry = R \\ 0, & \text{overige gevallen.} \end{cases}$$

Proces I

$\alpha$	$\beta$	R(10)	R(20)	R(50)
0	0	.34	.32	.31
0	1	.33	.33	.31
0	2	--	--	--
-.25	0	.31	.35	.32
-.5	0	.25	.30	.30
-.75	0	--	--	--

Proces II

-.5	-.5	.27	.32	.31
0	0	.31	.34	.31
0	1	.32	.33	.31
0	2	.30	.28	.26
0	3	.22	.19	.21
-.25	0	.31	.33	.31
-.5	0	.29	.31	.31
-.75	0	.25	.28	.30
.5	-.5	.34	.32	.30

Voor twee gevallen stelden we, in verband met lage waarden van  $R(50)$ , het aantal iteraties op 150. Ter vergelijking zijn de resultaten van het Richardson-Chebyshev proces ernaast afgedrukt.

#### Proces II

start- vector	$\alpha$	$\beta$	$R(10)$	$R(20)$	$R(50)$	$R(100)$	$R(150)$
5	0	3	.34	.26	.24	.25	.26
5	-.5	-.5	.23	.30	.32	.32	.32
6	0	3	.22	.19	.21	.24	.17
6	-.5	-.5	.27	.32	.31	.25	.17

#### 5.2. Effect van de gewichtsfunctie

Door keuze van de gewichtsfunctie kan men bepalen welke delen van het spectrum van  $H$  bij het iteratieproces benadrukt zullen worden. De onderzochte mogelijkheden zijn:

$$\text{proces I : } w = (1-y)^\alpha (1+y)^\beta$$

$$\text{proces II: } w = (1-y)^\alpha (1+y)^\beta (y_1 - y)^{-1} \quad \alpha, \beta > -1.$$

In het door ons berekende geval ( $R = 10$ ) is  $y_1 = 1.05$ .

Variatie van  $\alpha$  en  $\beta$  heeft bij beide processen hetzelfde effect: naarmate  $\alpha$  stijgt en/of  $\beta$  daalt wordt de omgeving van  $y = 1$  meer beklemtoond en die van  $y = +1$  meer verwaarloosd (en andersom).

We zullen nagaan of dit effect ook blijkt uit de waarnemingen:

Startvector 3 is de eigenvector van  $H$  met de grootst mogelijke eigenwaarde; iedere beklemtoning van de omgeving van  $y = 1$  vergroot de convergentiesnelheid.

Startvector 4 en 5 hebben vergeleken met startvector 3 steeds lagere eigenwaarden in hun spectrum, het spectrum wordt breder. Bij startvector 5, proces II,  $\alpha = 0$ ,  $\beta = 3$  ziet men een geval van verwaarlozing van het lage deel van het spectrum.

Startvector 6 is moeilijk vergelijkbaar met de vorige startvectoren, de beginfout heeft een breed spectrum. Het geval, proces II,  $\alpha = 0$ ,  $\beta = 3$ , is weer een voorbeeld van verwaarlozing van het lage deel van het spectrum.

### 5.3. Conclusies

Door variëren van de gewichtsfunctie kan, afhankelijk van de startvector, een flinke variatie in de convergentiesnelheid worden verkregen. Deze variatie is vaak toe te schrijven aan slechte keuze van de gewichtsfunctie (verwaarlozing van een deel van het spectrum).

Convergentieversnellingen ten opzichte van het Richardson-Chebyshev-proces zijn het duidelijkst waarneembaar bij een klein aantal iteraties. Proces I heeft, bij een groot aantal iteraties, dezelfde theoretische convergentiesnelheid als het Richardson-Chebyshev-proces (ieder in de bijbehorende norm).

Het lijkt niet eenvoudig om bij een bepaalde startvector een gewichtsfunctie te vinden die een proces levert dat zeker sneller is dan het Richardson-Chebyshev-proces.

Men krijgt enigszins de indruk dat "A plateau is reached, however, beyond which it is difficult to envision significant gains in efficiency without a fresh approach" ([3], pag. 270).

### 6. ALGOL 60 programma voor proces II

Hieronder volgt een ALGOL 60 programma voor proces II, bestemd voor het MC-ALGOL 60-systeem voor de X8 [4]. Dit proces wordt beschreven door (1.9), (3.2), (4.5) en (4.6). Bij iedere iteratiestap wordt de convergentiesnelheid (5.1) afgedrukt. Bij de laatste iteratie wordt de oplossing afgedrukt.

De getalband moet achtereenvolgens bevatten:

R	aantal roosterpunten
K	aantal iteraties
i	gewenste startvector
$\alpha$	gewenste gewichtsfunctie (4.4)
$\beta$	gewenste gewichtsfunctie (4.4)

doorgaan als doorgaan = -1 eindigt het programma na de K-de iteratie;  
als doorgaan = 1 kan men weer een nieuwe rij getallen, R, K,  
i, etc. in laten lezen.

Het programma is voorzien van comments; voor de identifiers zijn dezelfde letters gebruikt als in dit verslag.

```

begin comment mjwj,21 1 71 ,opdrnr R2082,
  richardson-jacobi methode voor model-dirichlet probleem,
  de matrix is die van een jacobi proces met plusformule;
  integer K,R,Rmineen,k,rx,ry,i;
  real alfa,beta,
    mumin,mumax,
    pim,pin,pip,
    y1,ha,eps,
    Dn,d,lDnul,Rn,
    an,bn,cn,noemern;
opnieuw:
  R:=READ;Rmineen:=R-1;
  K:=READ;
  alfa:=READ;beta:=READ;
  i:=READ;
  ha:=3.141592653589793238462643/R;
  mumin := -1+.5xha xha;mumax:=1-.5xha xha;
  y1:=(2-mumax-mumin)/(mumax-mumin);
  eps:=y1-1;
  begin
    comment de achtervoegsels m,n en p geven aan hoever een
    identifier onder of boven het rekursieniveau ligt,
    m betekent min,n betekent niveau en p betekent plus;
    real array um[0:R,0:R],un[0:R,0:R],up[0:R,0:R],yn[0:R,0:R];
    real procedure F(i);integer i;
    begin comment F(i) bepaalt beginapproximatie;
      if i=1 then F:=READ else
      if i =2 then F:=0 else
      if i=3 then F:=sin(rx xha)xsin(ry xha) else
      if i=4 then F:=(rx xha-2)x(ry xha-2)xsin(rx xha)xsin(ry xha) else
      if i=5 then F:=(rx xha-2)x(ry xha-2)x(rx xha-1)x(ry xha-1)x
        sin(rx xha)xsin(ry xha) else
      if i=6 then F:= if ry=R then 1 else 0
    end;
  end;

```

```

procedure YH;
begin comment YHberekent y(H) un ;
  for rx:=1 step 1 until Rmineen do
    for ry:=1 step 1 until Rmineen do
      yn[rx,ry]:=(-mumin-mumax)/(mumax-mumin)×un[rx,ry]+
        2/(mumax-mumin)×.25×
        (un[rx,ry-1]+un[rx-1,ry]+un[rx+1,ry]+un[rx,ry+1]);
    for rx:=0 step 1 until R do
      begin yn[0,rx]:=y1×un[0,rx];
        yn[R,rx]:=y1×un[R,rx];
        yn[rx,0]:=y1×un[rx,0];
        yn[rx,R]:=y1×un[rx,R]
      end
    end;
end;
procedure start;
begin comment wij starten op recursieniveau k is 0
  pi is jacpol in y1,u is oplossing;
  pin:=1;
  pip:=.5×(alfa-beta)+.5×(alfa+beta+2)×y1;
  for rx:=0 step 1 until R do
    for ry:=0 step 1 until R do
      un[rx,ry]:=F(i);
    end;
  YH;
  for rx:=0 step 1 until R do
    for ry:=0 step 1 until R do
      up[rx,ry]:= (.5×(alfa-beta)×un[rx,ry]+
        .5×(alfa+beta+2)×yn[rx,ry])/pip;
    end;
  end;
procedure recurreer;
begin comment berekent recursief de nieuwe waarden van un;
  noemern:=(2×(k+1)×(k+alfa+beta+1)×(2×k+alfa+beta));
  an:=(2×k+alfa+beta+1)×(alfa×alfa-beta×beta)/noemern;
  bn:=(2×k+alfa+beta+2)×(2×k+alfa+beta+1)×
    (2×k+alfa+beta)/noemern;
  cn:=2×(k+alfa)×(k+beta)×(2×k+alfa+beta+2)/noemern;
  pim:=pin;pin:=pip;pip:=(an+bn×y1)×pin-cn×pim;
  for rx:=0 step 1 until R do
    for ry:=0 step 1 until R do
      begin um[rx,ry]:=un[rx,ry];
        un[rx,ry]:=up[rx,ry]
      end;
    end;
  YH;
  for rx:=0 step 1 until R do
    for ry:=0 step 1 until R do
      up[rx,ry]:= an×pin/pip×un[rx,ry]+
        bn×pin/pip×yn[rx,ry]-
        cn×pim/pip×um[rx,ry];
    end;
  end;
end;

```

```

procedure Dun;
begin comment Dun berekent (1-H)un;
  Dn:=0;
  for rx:=1 step 1 until Rmineen do
  for ry:=1 step 1 until Rmineen do
  begin d:=un[rx,ry]-.25*(un[rx,ry-1]+un[rx-1,ry]+
    un[rx+1,ry]+un[rx,ry+1]);
    Dn:=if Dn>abs(d) then Dn else abs(d);
  end
end;
procedure output;
begin comment drukt un af;
  integer s;
  NEWPAGE;
  NLCR;NLCR;PRINTTEXT(† oplossing na †);
  ABSFIXT(3,0,k-1);PRINTTEXT(† iteraties †);NLCR;
  if R< 12 then s:=1 else if R<24 then s:=2 else
  if R< 36 then s:=3 else if R< 48 then s:=4 else goto EIND;
  for ry:=0 step 1 until R do
  begin NLCR;for rx:=0 step 1 until R do
    FLOT(5,2,un[rx,ry])
  end;
  EIND:
end;
BEGINPROGRAMMA:
  NEW PAGE;
  NLCR;PRINT(R);PRINT(K);PRINT(eps);
  PRINT(alfa);PRINT(beta);
  PRINT(i);
  start;
  Dun;lDnul:=ln(Dn);
  for k:= 1 step 1 until K do
  begin recurrence;
    Dun;Rn:=-1/k*(ln(Dn)-lDnul);
    NLCR;PRINT(k);FLOT(5,2,pin);FLOT(5,2,Rn);
  end;
  output;
end;
if READ > 0 then goto opnieuw;
end

```

## Literatuur

- [1] P.J. van der Houwen, Numerieke oplossingsmethoden; Colloquium  
elliptische differentiaalvergelijkingen, hoofdstuk 6.  
MC Syllabus 9.2, Amsterdam 1970.
  
- [2] E.L. Stiefel, Kernel polynomials in linear algebra and their  
numerical applications; Further contributions to the  
solution of simultaneous linear equations and the deter-  
mination of eigenvalues.  
National bureau of standards, Applied mathematics series  
49, 1958.
  
- [3] E.L. Wachspress, Iterative solution of elliptic systems.  
Prentice-Hall, Inc., 1966.
  
- [4] F.E.J. Kruseman Aretz, Het MC-ALGOL 60-systeem voor de X8.  
Voorlopige programmeurshandleiding.  
MR 81, Mathematisch Centrum, Amsterdam, 1966.